Lógica

Sistemas formales y demostración

Damiano Zanardini

Graduado/a en Ingeniería Informática
Facultad de Informática
Universidad Politécnica de Madrid
damiano@fi.upm.es

Curso Académico 2010/2011

Consecuencia lógica

Consecuencia lógica

Dado un conjunto de fórmulas $\Gamma = \{F_1, .., F_n\}$ y una fórmula G sobre el mismo lenguaje, G es *consecuencia lógica* de Γ (escrito $\Gamma \models G$) sii toda valoración que satisface a Γ también satisface a G, o, alternativamente, no hay ninguna valoración que satisfaga a Γ y que no satisfaga a G

Nota

- satisfacer a un conjunto $\{F_1, .., F_n\}$ significa satisfacer a todas las fórmulas del conjunto

Consecuencia lógica

Ejemplo: $\{p \rightarrow (q \rightarrow r),\ p \wedge q\} \models r$

p	q	r	$p \rightarrow (q \rightarrow r)$	$p \wedge q$	$(p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)$
v	**v**	**v**	**v**	**v**	**v**
v	**v**	**f**	**f**	**v**	**f**
v	**f**	**v**	**v**	**f**	**f**
v	**f**	**f**	**v**	**f**	**f**
f	**v**	**v**	**v**	**f**	**f**
f	**v**	**f**	**v**	**f**	**f**
f	**f**	**v**	**v**	**f**	**f**
f	**f**	**f**	**v**	**f**	**f**

Consecuencia lógica

Ejemplo: $\{p \rightarrow (q \rightarrow r),\ p \wedge q\} \models \neg r$

p	q	r	$p \rightarrow (q \rightarrow r)$	$p \wedge q$	$(p \rightarrow (q \rightarrow r)) \wedge (p \wedge q)$	$\neg r$
v	**v**	**v**	**v**	**v**	**v**	**f**
v	**v**	**f**	**f**	**v**	**f**	**v**
v	**f**	**v**	**v**	**f**	**f**	**f**
v	**f**	**f**	**v**	**f**	**f**	**v**
f	**v**	**v**	**v**	**f**	**f**	**f**
f	**v**	**f**	**v**	**f**	**f**	**v**
f	**f**	**v**	**v**	**f**	**f**	**f**
f	**f**	**f**	**v**	**f**	**f**	**v**

Sintaxis vs. semántica

Sistemas formales

Un *sistema formal de demostración* consiste en:

- un lenguaje formal (alfabeto y reglas de formación de fórmulas)
- un conjunto de *axiomas lógicos* o *axiomas* (fórmulas válidas, sin prueba)
- un conjunto de *reglas de inferencia* para demostrar fórmulas: un cálculo
- una definición de demostración

Teorías

Una *teoría* T es un sistema formal ampliado con un conjunto Γ de *axiomas no lógicos* o *premisas* (es decir, que se consideran como verdad)

$$T[\Gamma]$$

- si $\Gamma = \emptyset$ entonces T es la *teoría básica* del sistema formal

Sintaxis vs. semántica

Demostraciones

Una *demostración* o *prueba* de una fórmula G en una teoría $T[\Gamma]$ (escrito $T[\Gamma] \vdash G$) es una secuencia *finita* de fórmulas tal que

- toda fórmula de la secuencia es
  - un axioma o premisa de la teoría; o
  - el resultado de la aplicación de una regla de inferencia a fórmulas anteriores en la secuencia

$$[\mathit{nombre\_de\_la\_regla}] \ \frac{\mathit{premisas}}{\mathit{conclusión}}$$

- G es la última fórmula de la secuencia

Teoremas (de una teoría)

- un *teorema* de la teoría $T[\Gamma]$ es una fórmula para la que existe al menos una prueba en $T[\Gamma]$

Lo que se pide a un sistema formal

Teorema (corrección (validez))

Todos los teoremas de $T[\Gamma]$ son consecuencias lógicas suyas:

$$\textit{si } T[\Gamma] \vdash G \textit{ entonces } \Gamma \models G$$

Teorema (completitud)

Dada una teoría $T[\Gamma]$, todas las consecuencias lógicas de Γ son teoremas de $T[\Gamma]$:

$$\textit{si } \Gamma \models G \textit{ entonces } T[\Gamma] \vdash G$$

- *durante el resto del curso hablaremos de cómo hallar estas fórmulas*

Nota

- si la completitud es muy importante, la corrección es indispensable

Un poco de metalógica

Hechos

- decir $\emptyset \models G$ es igual que decir que G es válida (no necesita de premisas)
- si el cálculo es correcto y completo, entonces $\models$ y $\vdash$ son equivalentes
  - este hecho es tan importante que el símbolo de la *Association of Logic Programming* es

Un poco de metalógica

Teorema (Deducción)

- $\{F_1, .., F_n\} \cup \Gamma \models G \quad sii \quad \Gamma \models (F_1 \wedge .. \wedge F_n) \rightarrow G$

Un poco de metalógica

Teorema (Deducción)

- $\{F_1, .., F_n\} \cup \Gamma \models G \quad sii \quad \Gamma \models (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $T[\{F_1, .., F_n\} \cup \Gamma] \vdash G \quad sii \quad T[\Gamma] \vdash (F_1 \wedge .. \wedge F_n) \rightarrow G$

Un poco de metalógica

Teorema (Deducción)

- $\{F_1, .., F_n\} \cup \Gamma \models G \quad sii \quad \Gamma \models (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $T[\{F_1, .., F_n\} \cup \Gamma] \vdash G \quad sii \quad T[\Gamma] \vdash (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $\{F_1, .., F_n\} \models G \quad sii \quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$

Un poco de metalógica

Teorema (Deducción)

- $\{F_1, .., F_n\} \cup \Gamma \models G \quad sii \quad \Gamma \models (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $T[\{F_1, .., F_n\} \cup \Gamma] \vdash G \quad sii \quad T[\Gamma] \vdash (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $\{F_1, .., F_n\} \models G \quad sii \quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$
- $T[\{F_1, .., F_n\}] \vdash G \quad sii \quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$

Un poco de metalógica

Teorema (Deducción)

- $\{F_1, .., F_n\} \cup \Gamma \models G \quad sii \quad \Gamma \models (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $T[\{F_1, .., F_n\} \cup \Gamma] \vdash G \quad sii \quad T[\Gamma] \vdash (F_1 \wedge .. \wedge F_n) \rightarrow G$
- $\{F_1, .., F_n\} \models G \quad sii \quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$
- $T[\{F_1, .., F_n\}] \vdash G \quad sii \quad VAL((F_1 \wedge .. \wedge F_n) \rightarrow G)$
- $T[\{F_1, .., F_n\}] \vdash G \quad sii \quad INSAT(F_1 \wedge .. \wedge F_n \wedge \neg G)$

Un poco de metalógica

Teorema (Intercambio)

Si

- $\vdash F_1 \leftrightarrow F_2$; *y*
- *G es una fórmula con un hueco (ej.* $p \wedge (q \vee \_)$*); y*
- $G[H]$ *es la fórmula que se obtiene tapando el hueco de G con H (ej.* $p \wedge (q \vee (r \rightarrow p))$ *si H es* $r \rightarrow p$*)*

entonces

$$\vdash G[F_1] \leftrightarrow G[F_2]$$

Nota

- deriva de un teorema correspondiente que usa $\models$
- las dos formas son equivalentes si $\vdash$ es correcto (válido) y completo